Developing an Efficient Feature Engineering and Machine Learning Model for Detecting IoT-Botnet Cyber Attacks

M.Tech Dissertation

To the Faculty of Engineering & Technology of

Mahatma Gandhi Mission University, Aurangabad -431003

In partial fulfillment of the award of the degree

of

Master of Technology

in

COMPUTER SCIENCE & ENGINEERING - DIGITAL TRANSFORMATION

by

Anuja Manik Lakhe
PRN No.202001110004

Under the Guidance of
Prof. S.N.Jaiswal

Department of Computer Science & Engineering

Jawaharlal Nehru Engineering College, Aurangabad

YEAR 2021-2022

MGM University

JAWAHARLAL NEHRU ENGINEERING COLLEGE

N-6, CIDCO, AURANGABAD (M.S.), INDIA.

CERTIFICATE

This is to certify that the Dissertation report entitled " **Developing an Efficient Feature Engineering and Machine Learning Model for Detecting IoT-Botnet Cyber Attacks**" has been submitted by Miss. **Anuja Manik Lakhe** with PRN No.**202001110004** in partial fulfillment for the award of the degree of Master of Technology in **COMPUTER SCIENCE & ENGINEERING - DIGITAL TRANSFORMATION** from Jawaharlal Nehru Engineering College, MGM University, Aurangabad.

The matter embodied in this dissertation report is a record of his/her own independent work carried out by him/her under my supervision and guidance. The matter embodied in this report has not been submitted to any other University or Institute for the award of any degree or diploma.

Guide	**H.O.D**	**Dean**
Prof S.N.Jaiswal	Dr V.B.Musande	Dr. H.H.Shinde
Professor	Professor& Head	(Engg. & Tech.)
CSE Dept	CSE Dept	MGMU

Dissertation Approval Certificate

Viva-voce examination for the above dissertation work is conducted on

_____ and the work is approved for the award Master of Technology in

COMPUTER SCIENCE & ENGINEERING - DIGITAL TRANSFORMATION

Internal Examiner **External Examiner**

UNDERTAKING

Icertify that

a. The work contained in the dissertation report is original and has been done by myself under thesupervisionof my guide.

b. Theworkhas notbeen submittedtoanyotherInstitutefor anydegree ordiploma.

c. WheneverIhaveusedmaterials(data,theoreticalanalysis,andtext)fromothersources, I have given due credit to them by citing them in the text of the dissertation reportandgivingtheirdetails inthe references.

d. From the plagiarism test, it is found that the similarity index of the dissertation reportislessthan or equal to 10% which is permissible as perthe UGC (Promotion of Academic Integrity and Prevention of Plagiarism in Higher Educational Institutions) Regulations, 2018.

Date:

Place: Aurangabad

Name of Student: Anuja Manik Lakhe **Sign:**

PRN No.202001110004

ACKNOWLEDGEMENTS

It is a great sense of satisfaction that I represent my real venture on practical computing as project work. As the curriculum requirement of the course of M.TECH. Computer, I have successfully completed project work **Developing an Efficient Feature Engineering and Machine Learning Model for Detecting IoT-Botnet Cyber Attacks Using CNN and ML Algorithms**, under the guidance of **Prof. S.N.Jaiswal** and it's my first duty to express my hearty gratitude to him and to all those personalities who help me to partially complete this project work easily and successfully.

I offer my sincere thanks to **Prof. S.N.Jaiswal**, who guided me at every stage with valuable suggestions and guidelines while planning and designing the project. I am also thankful to all staff members of the Computer Science and Engineering department, for their suggestions and moral support in this project.

I would like to thank **Dr. V.B.Musande**, Head of the Computer Science and Engineering department, for her unwavering support during this project work. Thank You!!

Anuja Manik Lakhe
PRN.No. 202001110004
M.Tech Computer Science and Engineering

ABSTRACT

Detection of Botnet attacks is one of the most important and challenging tasks in today's era, everything from educational institutions to the corporate world is shifted from offline to online. It leads to an exponential increase in intrusions and attacks over Internet-based technologies. Additionally, it is difficult for existing security systems to detect intrusion because attackers continuously change their strategies. As a result of the great degree of diversity and similarity between attacks, it is difficult to identify attacks. In this study, we put forth a technique for using a convolutional neural network and Machine learning algorithms to detect distributed denial of service attacks. When these algorithms are applied to the data the prediction of attacks is done very fast and with higher accuracy. These predictions help the legitimate user. In this proposed work different Machine learning and deep learning algorithms are used such as convolution neural network, long short-term memory RNN, Naïve Bayes, K-nearest neighbor, and Random Forest. The goal of this project is to design and create a system that accurately detects DDoS attacks. It is observed that CNN results gave the highest accuracy of 92%, precision of 90%, and recall of 93% over other machine learning algorithms.

Table of Contents

ABSTRACT	iii
NOMENCLATURE	v
LIST OF TABLES	v
LIST OF FIGURES	vi
1. INTRODUCTION	1
1.1 Overview	1
1.2 Application	2
2. LITERATURE SURVEY	3
3. PROBLEM DEFINITION AND SCOPE	6
3.1 Problem Statement	6
3.2 Objectives	6
3.3 Motivation	7
4. SYSTEM DEVELOPMENT	8
4.1 Proposed System	8
4.2 Algorithms and Mathematical Model	9
5. HARDWARE / SOFTWARE DESCRIPTION	16
5.1 Software Requirement Specification	16
5.2 Hardware Specification	16
6. EXPERIMENTATION AND RESULTS	17
6.1 Dataset Used	17
6.2 Performance Measure	17
6.3 Results and Discussion	19
7. CONCLUSION AND FUTURE SCOPE	23
8. REFERENCES	24
9. ANNEXURE A: Paper Publication	25
10. ANNEXURE B : Presentation	35
11. ANNEXURE D: Similarity Check	38

NOMENCLATURE

ABBREVIATION	EXPANSION
CNN	Convolution Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
RF	Random Forest
NB	Naïve Bayes
KNN	K-Nearest Neighbor

LIST OF TABLES

TABLE	TABLE TITLE	PAGE NO.
1	Literature Survey	3
3	Comparative Analysis of Classifiers	21

LIST OF FIGURES

FIGURE NO.	ILLUSTRATION NAME	PAGE NO.
1	Proposed System	8
2	Feature Extraction and Classification using CNN	10
3	Example of Convolution	10

4	Example of RELU	11
5	Example of Max Pool	12
6	Flatten Layer	12
7	Complete CNN architecture	13
8	Random Forest Algorithm for botnet attack prediction	13
9	KNN for botnet attack prediction	15
10	Confusion Matrix	18
11	CNN+ LSTM Training	19
12	CNN+ LSTM Testing	19
13	ML Algorithms	20
14	CNN+ LSTM Confusion Matrix	20
15	Comparison of accuracy of classifiers	22

Chapter 1

1. INTRODUCTION

1.1 Overview

The most common threat to online service providers is a distributed denial of service (DDoS) attack. It involves the attacker's ability to compromise the availability of web services offered by the targeted host. This is achieved by using attacking agents such as botnet and or compromised Internet of Things (IoT) devices to exhaust the victim's computing capacity (Network Bandwidth, System and Application resources) preventing service availability to legitimate users.

According to a few authors, the main victims of DDoS attacks are organizations with an online presence. The effect of DDoS attacks on these organizations ranges from very simple problems to significant ones such as financial losses, compromise of national security, and endangerment of human life. Distributed Denial of Service (DDoS) IoT Botnet Attack is one of the most widespread securities. IoT Botnet attacks to internet service providers. It is the most easily launched attack, but very difficult and expensive todetect and mitigate. In view of the devastating effect of the DDoS IoT Botnet Attack, there has been an increase in the adaptation of a network detection technique to reveal the presence of DDoS attacks before huge traffic buildup to prevent service availability.

Several works done on DDoS attack detection reveals that the conventional DDoS attack detection methods based on statistical divergence are useful, however, the large surface area of the internetwhich serve as the main conduit for IoT BotnetAttack to occur makes it difficult to use thisapproach to detect IoT Botnet Attack on the network. Hence this research work is focused on using detectiontechniques based on a deep learning technique because it is proven as the most effective detectiontechnique against DDoS IoT Botnet attacks.

Out of the several deep neural network techniques available, this research focuses on one aspectof recurrent neural network called Long Short-Term Memory (LSTM) and the TensorFlow framework to build and train a deep neural network model to detect the presence of DDoS IoT Botnet Attack on a network. This model can be used to develop an Intrusion Detection System (IDS) to aid in detecting DDoS IoT Botnet Attacks on the network.

Also, at the completion of this project, the expectation of the produced model is to have a higher detection accuracy rate and a low false alarm rate. Design Science Research Methodology (DSRM) was used to carry out this project. The test experiment for this work was performed on CPU and GPU base systems to determine the base system's effect on the detection accuracy of the model.

Seven evaluating parameters were used to test the model detection accuracy and performance on both Central Processing Unit (CPU) and Graphics Processing Unit (GPU) systems to achieve the set goals.

1.2 Application

Bot attack detection is key to catching bots and preventing them from having a negative or bad effect on business. Detection systems are good for catching malicious activities and protecting data from hacking. It also provides protection from data breaches and improves business performance. Bots detection should be used to enhance customer experience. A detection system avoids the risk to many online applications for example E-commerce businesses. Bot detection system secures online services and affects the reputation of service providers. It also allows us to stay ahead of attackers

- Botnet attacks becoming more frequent and severe
- Early detection of attacks may reduce risk
- Availability of service to legitimate users.
- This system will detect the various attacks based on a different dataset.
- This system uses LSTM and CNN to classify the attacks.

Chapter 2

2. LITERATURE SURVEY

Sr.No	Title	Strength	Algorithms/ Techniques
1.	Developing an Efficient Feature Engineering and Machine Learning Model for Detecting IoT-Botnet Cyber Attacks by Mrutyunjaya Panda, Abd Allah A. Mousa, And Aboul Ella Hassanien [2021, IEEE]	Efficient machine learning and deep learning with suitable feature engineering are suggested to detect and protect the network from such vulnerabilities in the future	Proposed different techniques to detect Botnet attacks. For feature engineering and decision making, they used K-medoid, Scatter search, decision tree, naïve Bayes, CNN
2.	Machine Learning Based Intrusion Detection for IoT Botnet by SikhaBagui, Xiao Jian Wang, And Subhash Bagui [2021, International Journal of Machine Learning and Computing]	A Bayes algorithm comparison with other methods that use the data divided into 10 attacks carried by 2 botnets, gafgyt and mirai.	The author used Z-score for the normalization of data. Logistic regression, support vector machine, and random forest for training and prediction.
3.	Deep learning approaches for detecting DDoS attacks: a systematic review by Meenakshi Mittal, Krishan Kumar, Sunny Behal [2021,Springer]	CNN-LSTM model has an accuracy greater than other models.	This paper discusses various pre-processing techniques and also reviewed deep learning techniques such as ANN, and CNN for the prediction of DDoS attacks.

4.	Botnet Attack Detection by Using CNN-LSTM Model for Internet of Things Applications by Hasan Alkahtani, Theyazn H. H. Aldhyani [2021, Hindawi Security and Communication Networks]	For detection attacks, BASHLITE and Mirai, on nine commercial IoT devices are used. Extracted data includes benign and malicious patterns. With various IoT devices CNN-LSTM model provides optimal accuracy.	The author used deep learning, a convolutional neural network, and long short-term memory (CNN-LSTM) algorithm.
5.	Machine Learning-Based IoT-Botnet Attack Detection with Sequential Architecture by Yan NaungSoeYaokai Feng,Paulus InsapSantosa,Rudy HartantoandKouichi Sakurai[2020, Sensors]	The defined system's lightweight nature and the best detection accuracy by applying the feature selection approach. Hybrid classification could also greatly assist in achieving the best rapid detection accuracy.	Proposed artificial neural network (ANN), Decision tree, and Naïve Bayes for prediction of DDoS attacks
6.	Network Flow-based IoT Botnet Attack Detection using Deep Learning by Sriram S, Vinayakumar R, Mamoun Alazab, Soman KP [2020, IEEE]	ML algorithms are implemented using Scikit-learn and DL models are implemented using TensorFlow with Keras.	This paper used various ML algorithms like Naïve Bayes, Random Forest, K nearest neighbor, Support vector Machine
7.	Detecting Denial of Service Attacks with Bayesian Classifiers and the Random Neural Network.by G. Oke, G. Loukas and E. Gelenbe[IEEE]	DDoS attacks were based on three main factors: the study of statistical features of incoming DDoS attack traffic.	This paper used various ML algorithms like Recurrent Random Neural Networks, Network Security, Intrusion Detection, Bayesian Decision Taking

The concerns shown and contributions made by the various research groups have therefore attracted the attention of both the industry and academia sectors to come up with a detection technique or mechanism which will serve as the first line of defense

against DDoS attacks mitigation. This is because the research groups all support the fact that online services are under attack and therefore need an effective and efficient system to help detect the presence of DDoS attacks on the network infrastructure so that the right mitigation techniques are applied before its devastating effects are experienced by users

In view of the line of discussion, this research work focuses on using a Deep learning technique called Long Short-Term Memory (LSTM) Recurrent Neural Networks (RNN) to develop and train a tensor artificial intelligence (AI) model which will detect the presence of IoT Botnet attack traffic patterns on the network, and achieve a high detection accuracy, and a low false alarm rate.

LSTM RNN was chosen as the technique for this work because, among the family of RNN techniques and existing conventional machine learning techniques, LSTM RNN is rated as the best.

Denial of service(DoS) attacks are immensely easy to dispatch, but the handling of such types of attacks is very difficult. The three generic stages i.e., Detection, Classification, and Response are used to create a system for the detection of DoS attacks. In DoS detection mainly we are analyzing the incoming traffic which is the same as a pattern recognition problem, for this various machine learning techniques have been used. In building the effect detection system author used a Bayesian classifier for calculating likelihood ratios of the existence of attacks, which acts as a statistical preprocessing method for extracting the features, and these features are combinedly used in Recurrent Random Neural Network (r-RNN) for making a comprehensive decision regarding the attack detection. RNNs function the same as the biophysical neural network as it is a biologically influenced framework, where signals are not analog signals. So, this RNN structure we are using for analyzing real-time Statistical data also differentiates traffic during the DoS attack

Chapter 3

3. PROBLEM STATEMENT AND OBJECTIVE

3.1 Problem Statement

To design and develop a system for the detection of botnet attacks using Convolutional Neural Networks, Long Short-Term Memory Recurrent Neural Networks, and ML algorithms to achieve a better accuracy model.

3.2 Objectives

1. To analyze data and its parameters to check any redundancy in data values that may affect prediction results.
2. To remove all empty/uncertainty values
3. Use different types of classifiers to compare the model accuracy in detecting the DDoS attacks
4. To build a hybrid model which provides better accuracy when compared with other models

3.3 Motivation

With increasing technological evolution in the Internet of Things (IoT), it has become crucial to intensify its security aspect. The attackers update their skills continuously and hence the existing detection mechanisms cannot identify the attack. Since the volume of data generated and stored has increased manifolds, the traditional detection mechanisms are not appropriate for detecting novel DDoS attacks.

1. More and more IoT devices are connecting to the internet
2. Botnet attacks becoming more frequent and severe

3. Early detection of attacks may reduce the risk
4. Availability of service to legitimate users.

Chapter 4

4.1 PROPOSED SYSTEM

This is the proposed system diagram for Botnet attack detection using CNN and different ML algorithms.

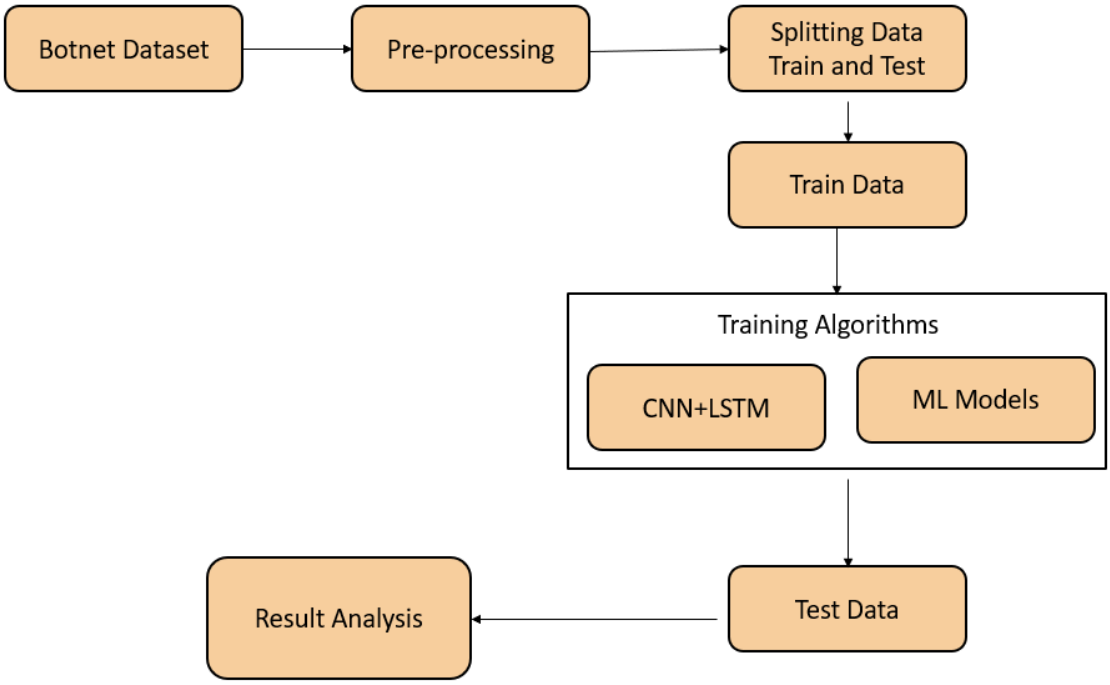


Fig 1: Proposed System

Data Collection: The first step is to collect data. We will collect the NSL-KDD dataset. A large amount of internet traffic is collected and stored in the form of a dataset. This contains attack and normal records.

Data Preprocessing: In this step, we will clean the data. We will manipulate or drop the data before it is used to ensure or enhance performance.

Feature Extraction: The aim of this step is to reduce the number of features in a dataset by creating new features from the existing ones (and then discarding the original features).

Splitting Dataset: The Dataset is split into Training Dataset and Test Dataset. Machine Learning Algorithm: Using the training dataset train the model with various machine learning methods like Bayesian classifier, support vector machine, neural network, etc. Result Analysis: Applying the test dataset to the model or the trained system for analyzing the result using various parameters such as Accuracy, Precision, and Recall.

4.2 Algorithms and Mathematical Model

Botnet attack detection is one of the required mechanisms to protect against malicious behavior in an IoT environment. Most of the detection systems are signature-based. It is difficult to detect new kinds of attacks. Because new attacks are just the variances of the known attacks. So, machine learning-based techniques have the ability to detect such types of attack variances. In this study, we applied popular machine-learning methodologies, such as Convolutional Neural Networks, Long -Short term memory Recurrent Neural Networks, Random Forest, Naïve Bayes, and K-nearest neighbor.

4.2.1 CNN

- Convolutional Neural Networks (which are additionally called CNN/ConvNets) is a kind of Artificial Neural Network that are known to be tremendously strong in the field of distinguishing proof just as picture order.

Four main operations in Convolutional Neural Networks are:

1. Convolution
2. RELU
3. Pooling
4. Fully connected layer

4.2.1.1 Convolution Layer

Convolution is the first layer to extract features from an input image.

Convolution preserves the relationship between pixels by learning image features using small squares of input data.

It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel.

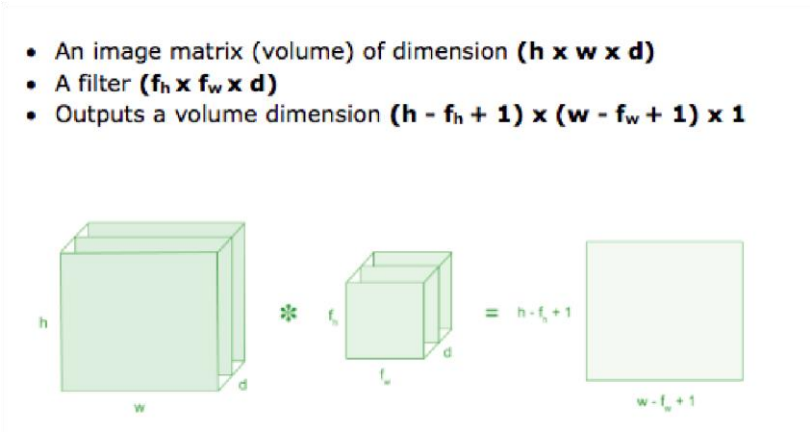


Fig 2: Feature Extraction and Classification using CNN

Example for Convolution

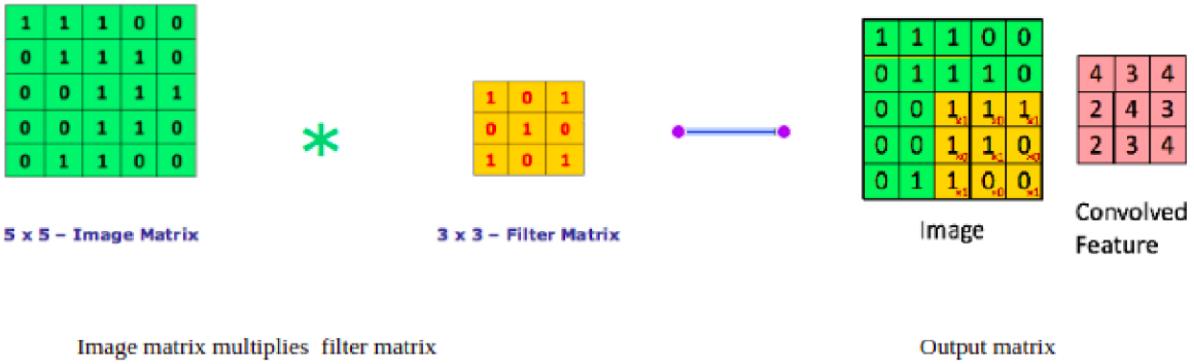


Fig 3: Example of Convolution

4.2.1.2 RELU- Rectified Linear Unit

- A piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero.

  The output is ***f(x) = max(0,x).***

Why ReLU is important :

ReLU's purpose is to introduce non-linearity in our ConvNet. Since, the real world data would want our ConvNet to learn would be non-negative linear values.

Example of RELU- Rectified Linear Unit

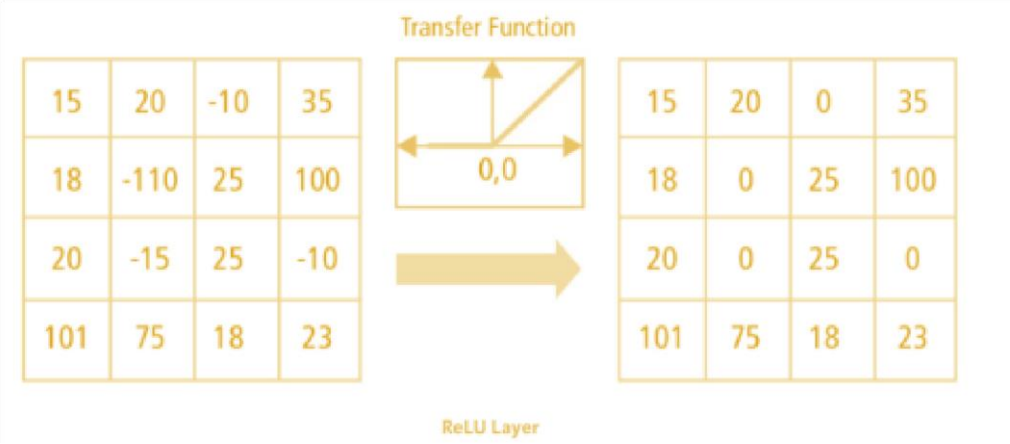


Fig 4 : Example of RELU

There are other nonlinear functions such as tanh or sigmoid that can also be used instead of ReLU. Most of the data scientists use ReLU since performance wise ReLU is better than the other two.

4.2.1.3 Pooling Layer

Pooling layers section would reduce the number of parameters when the images are too large.

Pooling can be of different types:

1. Max Pooling - takes the largest element from the rectified feature map.
2. Average Pooling -Taking the largest element could also take the average pooling.
3. Sum Pooling - Sum of all elements in the feature map call as sum pooling.

Example of Pooling Layer

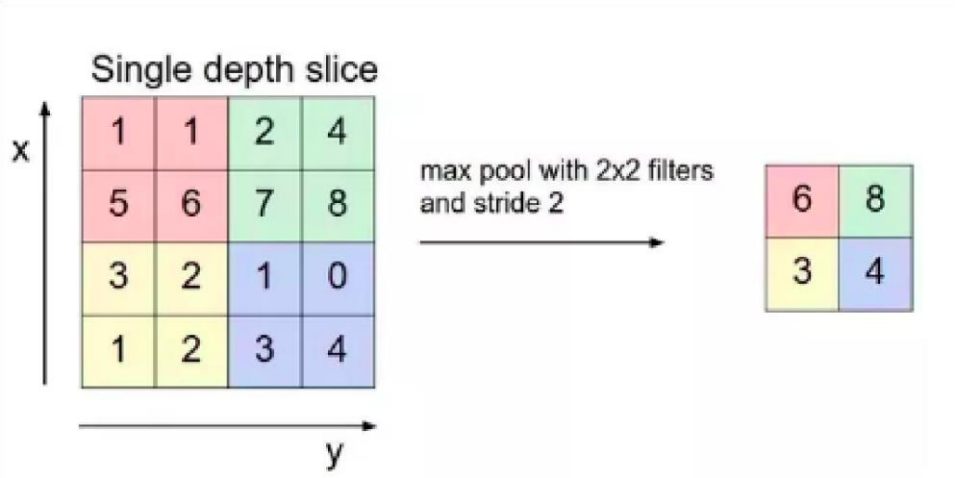


Fig 5: Example of Max Pool

4.2.1.4 Fully Connected Layer

Flattened matrix into the vector and feed it into a fully connected layer like a neural network.

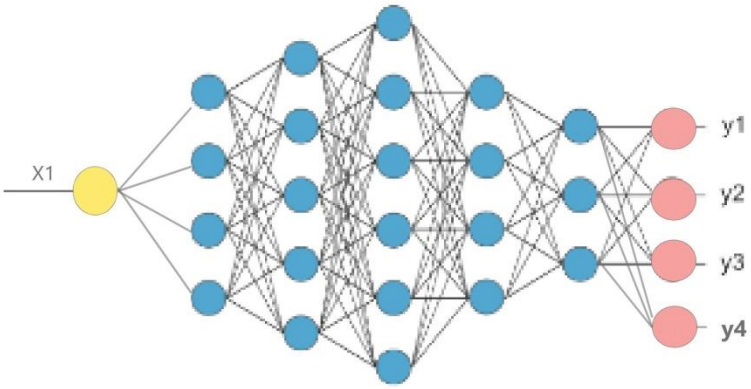


Fig 6: Flatten Layer

matrix is converted as vector (x1, x2, x3, …).

With the fully connected layers, combined these features together to create a model.

Finally, have an activation function such as softmax or sigmoid to classify the outputs.

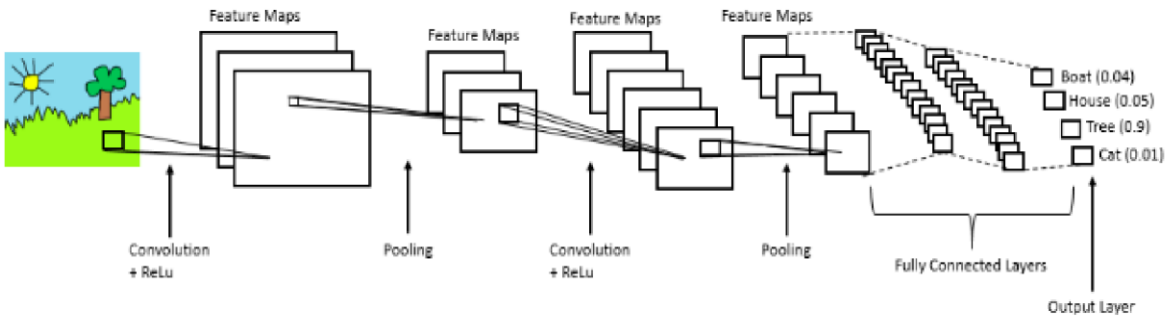


Fig 7: Complete CNN architecture

4.2.3 Random Forest

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.

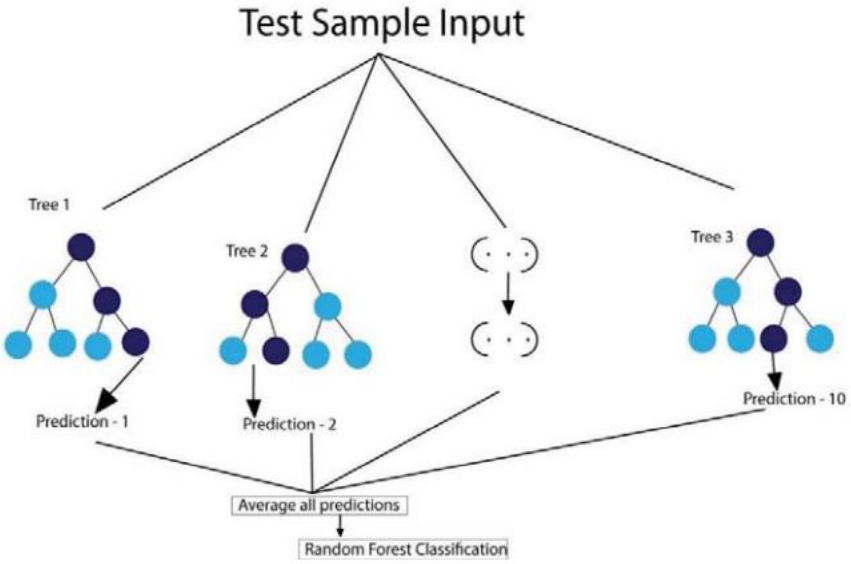


Fig 8 : Random Forest Algorithm for Attack Detection

4.2.5 Naive Bayes

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

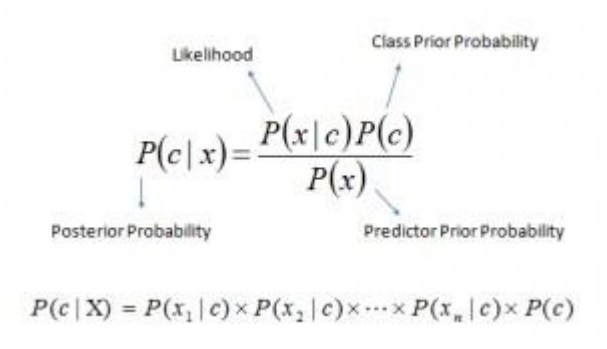
For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'.

The Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and

P(x|c).

Look at the equation below:


Likelihood
Class Prior Probability
$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$
Posterior Probability
Predictor Prior Probability
$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$

Above,

P(c|x) is the posterior probability of class (c, target) given predictor (x, attributes).

P(c) is the prior probability of class.

P(x|c) is the likelihood which is the probability of predictor given class.

P(x) is the prior probability of predictor.

4.2.7 K-Nearest Neighbor

K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on the Supervised Learning technique.

K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

This algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a good suite category by using K- NN algorithm.

It can be used for Regression as well as for Classification but mostly it is used for Classification problems, it does not make any assumption on underlying data.

It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.Training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

The K-NN working can be explained on the basis of the below algorithm:

Step-1: Select the number K of the neighbors

Step-2: Calculate the Euclidean distance of K number of neighbors

Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

Step-4: Among these k neighbors, count the number of the data points in each category.

Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.

Step-6: Our model is ready.

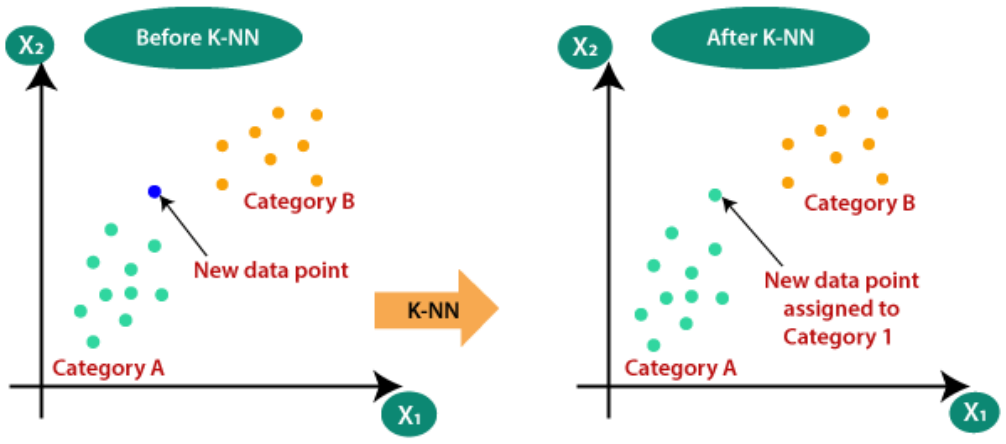


Fig 9 : KNN for Attack Prediction

Chapter 5

PROJECT DESIGN

5.1 Software Requirement Specification

- python 3.7.0

5.1 Hardware Specification

- Modern Operating System:
  - Windows 10
- x86 64-bit CPU (Intel architecture)
- 4 GB RAM

Chapter 6

EXPERIMENTATION AND RESULTS

Dataset Used

1. NSL-KDD Dataset
2. It has limited redundant records, so the classifiers will not be biased toward more frequent records.
3. There are fewer duplicate records; therefore, the performance of the learners is not biased by the methods which have better detection rates on the frequent records.
4. Dataset Features

    4.1 Total number of features: 41

    4.2 Number of numerical features: 38

    4.3 Number of categorical features: 3

Performance Measures

Accuracy

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations.

$$\text{Accuracy} = (TP + TN) / (TP + FP + TN + FN)$$

This term tells us how many right classifications were made out of all the classifications. In other words, how many TPs and TNs were done out of TP + TN + FP + FNs. It tells the ratio of "True"s to the sum of "True"s and "False"s.

Use case: Out of all Data that are tested using the system, how many were correctly classified.

Precision

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

$$\text{Precision} = TP / (TP + FP)$$

Out of all that were marked as positive, how many are actually truly positive.

Use case: Here, if there are attack features but marked as Normal , which is wrong prediction. Hence, the classification model which has the least FP value needs to be selected. In other words, a model that has the highest precision needs to be selected among all the models.

Recall / Sensitivity

Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes.

$$\text{Recall} = TP/ (TN + FN)$$

Out of all the actual real positive cases, how many were identified as positive.

Use case: Out of all the attacks or normal data tested, how many were actually predicted as Normal data. Hence, the classification model which has the least FN value needs to be selected. In other words, a model that has the highest recall value needs to be selected among all the models.

This is the confusion matrix for four classes we used to classify the attack. As we are using four different classes so the confusion matrix is of the 4*4 matrix.

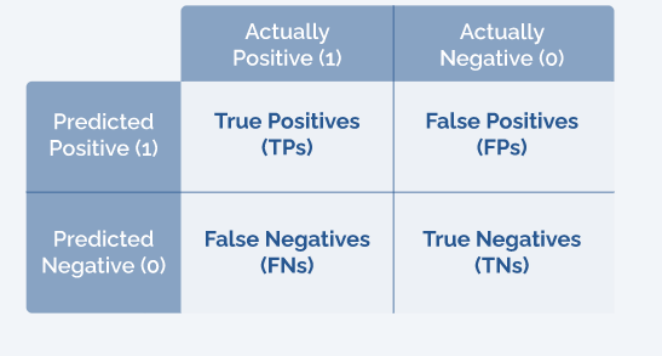


Fig 10: Confusion Matrix

Results and Discussions

Following are some snapshots of Botnet attack prediction.

1. CNN+LSTM model: The output shows the training of model using CNN +LSTM algorithm.

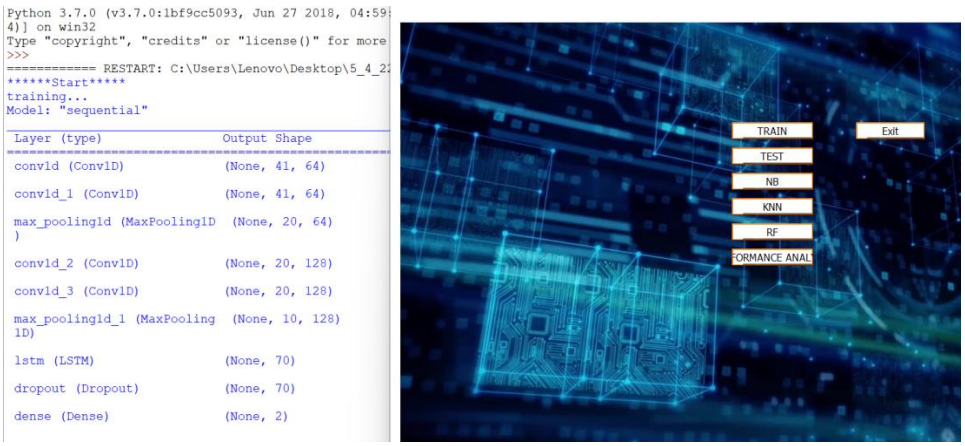


Fig 11. CNN+ LSTM Training

2. Testing: This output shows the different small sets of data using which we are trying to find the how accurately our model works.

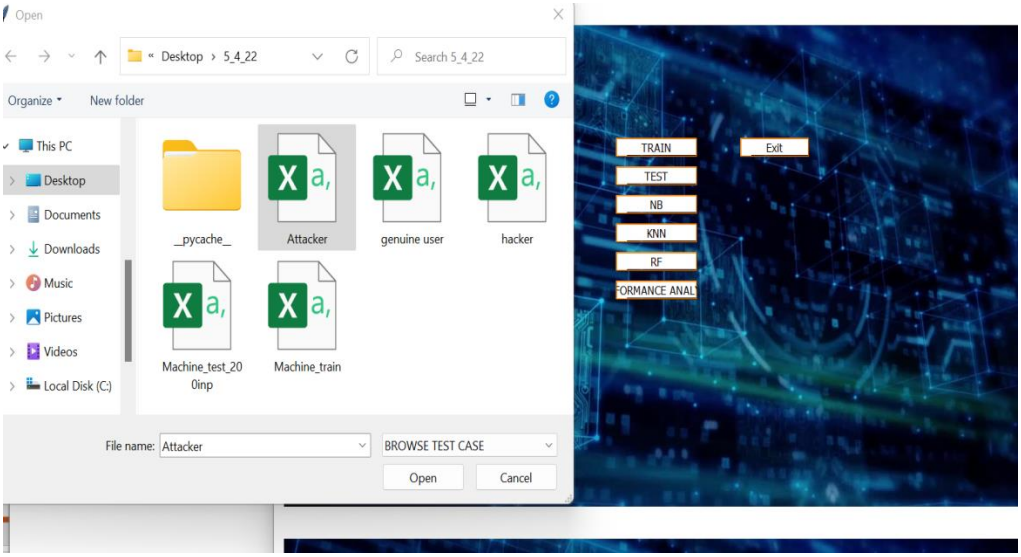


Fig 12. CNN+ LSTM Testing

3. ML Algorithms: This output shows the accuracy of model using different ML algorithms such as NB,KNN,RF.

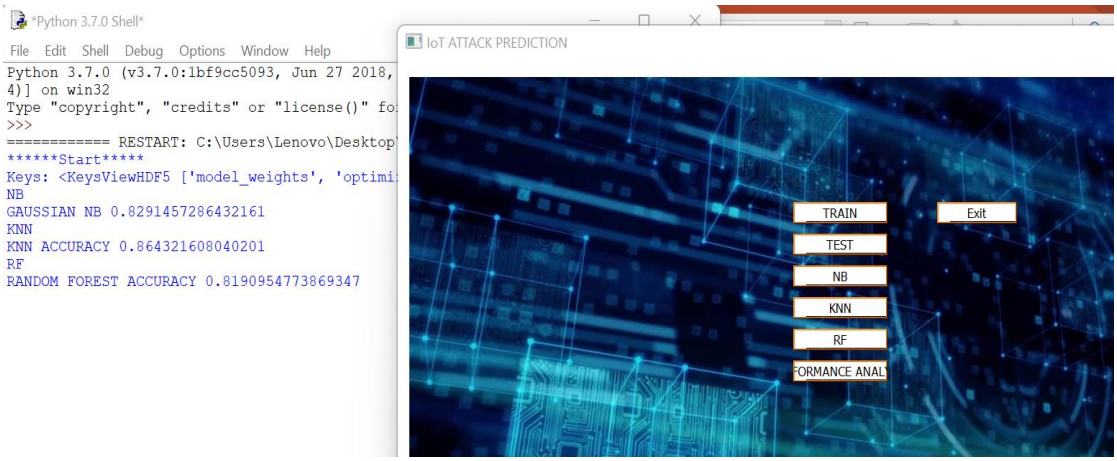


Fig 13. ML Algorithms

4. Performance of CNN+LSTM model: This output shows the confusion matrix for the CNN+LSTM model which is better that other algorithm (NB,KNN,RF).

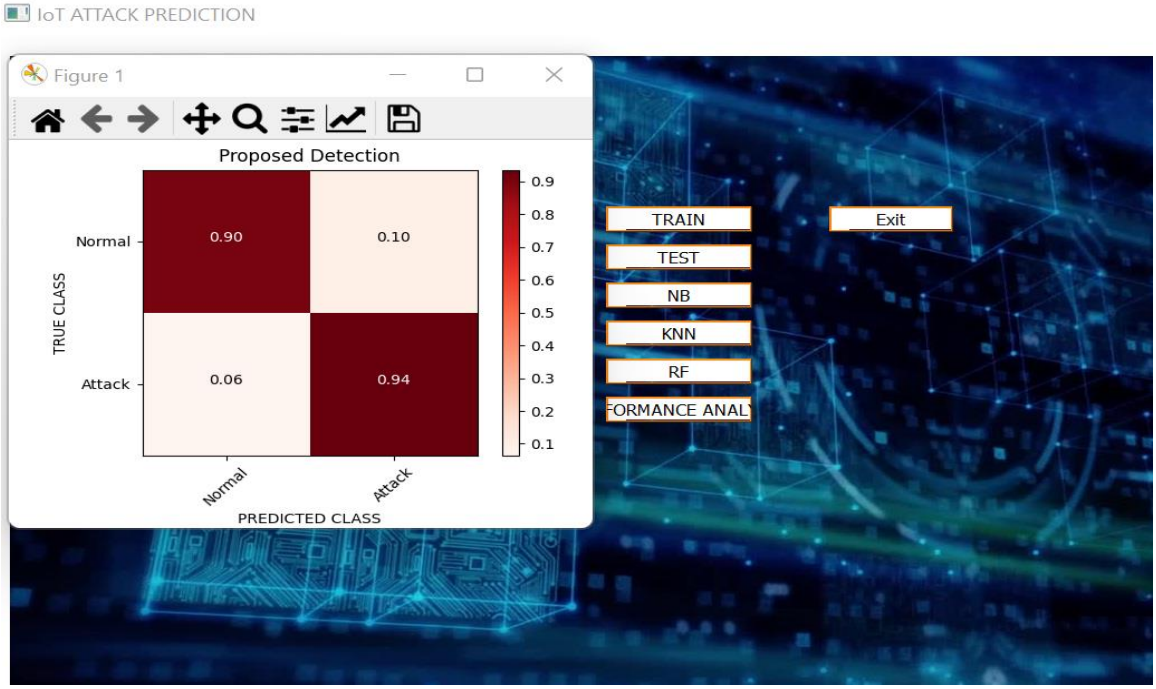


Fig 14. CNN+ LSTM Confusion Matrix

The following table shows the comparative analysis of the classification using different CNN and Machine Learning algorithms are as follows :

Comparative Analysis of Classifiers

Sr. No.	**Classification Algorithms**	**Accuracy**	**Precision**	**Recall**
1.	CNN+LSTM RNN	92.00	90.00	93.00
2.	Random Forest	81.90	79.71	82.98
3.	Naive Bayes	82.91	80.10	83.50
4.	K-nearest neighbor	86.43	85.88	84.77

The following line graph shows the comparative analysis of the classification using different CNN and Machine Learning algorithms are as follows :

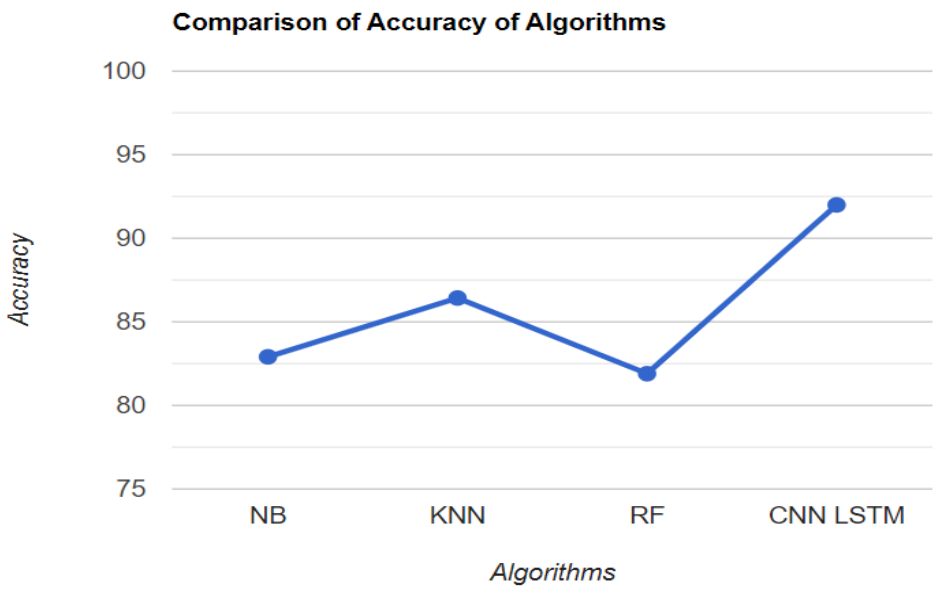


Fig 15: Comparison of accuracy of classifiers

Comments:

The experiment is conducted on 4 Machine Learning algorithms. It is observed that the accuracy of the K-nearest neighbor is 86.43% compared to other machine learning algorithms.The performance of this machine learning algorithm is compared with CNN. It is observed that the accuracy attack detection of CNN+LSTM RNN supersites all the machine learning algorithms i.e., 92.

Chapter 7

CONCLUSION

In this work, an IoT Botnet attack dataset like the NSL-KDD dataset was used for the prediction of DDoS attacks. The project aims to design a system that accurately detects attacks. To achieve these objectives, artificial intelligence, specifically, the Long Short-Term Memory (LSTM)Recurrent Neural Network (RNN), a machine learning technique, and Google's second-generation machine learning framework called TensorFlow were adopted and used

To determine the Accuracy and efficiency of the algorithm, it was evaluated against seven main parameters such as dataset size, epochs, learning rate, nodes, weights, and biases. There was a lot of iteration done to get the best results. Every iteration done involved assigning different values to parameters such as datasets size and epochs while randomly generated values for weights and biases were used, layers, nodes, and learning rate were held constant until the best detection accuracy was achieved.

During experimentation, it is observed that CNN+LSTM RNN results gave the highest accuracy of 92.00%, precision of 92.00%, and recall of 93.00% over other machine learning algorithms